

Pascal-XSC: Language Reference with Examples. By R. Klatte, U. Kulisch, M. Neaga, D. Ratz and Ch. Ullrich. Translated by G.F. Corless, R. Klatte, U. Kulisch, D. Ratz and C. Wolff. Springer-Verlag, Berlin, Heidelberg and New York. (1991, original; 1992, translation). 344 pages. \$42.00.

Contents:

1. Introduction. 1.1 Typography. 1.2 Historical remarks and motivation. 1.3 Advanced computer arithmetic. 1.4 Connection with programming languages. 1.5 Survey of PASCAL-XSC. 2. Language reference. 2.1 Basic symbols. 2.2 Identifiers. 2.3 Constants, types, and variables. 2.4 Expressions. 2.5 Statements. 2.6 Program structure. 2.7 Subroutines. 2.8 Modules. 2.9 String handling and text processing. 2.10 How to use dynamic arrays. 3. The arithmetic modules. 3.1 The module CLARI. 3.2 The module LARI. 3.3 The module CLARI. 3.4 The module MVARI. 3.5 The module MVCARI. 3.6 The module MVLARI. 3.7 The module MVCLARI. 3.8 The hierarchy of the arithmetic modules. 3.9 A complete sample program. 4. Problem-solving routines. 5. Exercises with solutions. 5.1 Test of representability. 5.2 Summation of exponential series. 5.3 Influence of rounding errors. 5.4 Scalar product. 5.5 Boothroyd/Dekker matrices. 5.6 Complex functions. 5.7 Surface area of a parallelepiped. 5.8 Parallelism and intersection of lines. 5.9 Transposed matrix, symmetry. 5.10 Rail route map. 5.11 Inventory lists. 5.12 Complex numbers and polar representation. 5.13 Complex division. 5.14 Electric circuit. 5.15 Alternating current measuring bridge. 5.16 Optical lens. 5.17 Interval evaluation of a polynomial. 5.18 Calculations for interval matrices. 5.19 Differentiation arithmetic. 5.20 Newton's method with automatic differentiation. 5.21 Measurement of time. 5.22 iterative method. 5.23 Trace of a product matrix. 5.24 Calculator for polynomials. 5.25 Interval Newton method. 5.26 Runge-Kutta method. 5.27 Rational arithmetic. 5.28 Evaluation of polynomials. A. Syntax diagrams. B. Indices and lists.

Unstructured Scientific Computation on Scalable Multiprocessors. Edited by P. Mehrotra, J. Saltz and R. Voigt. The MIT Press, Cambridge, Massachusetts and London, England. (1992). 405 pages. \$39.95.

Contents:

1. The Free-Lagrange method on the connection machine. (H.E. Trease and J.H. Cerutti). 2. Mapping unstructured grid problems to the connection machine. (S.W. Hammond and R. Schreiber). 3. Parallel unstructured grid generation. (R. Löhner, J. Camberos and M. Merriam). 4. Unstructured three-dimensional finite element simulations on data parallel architectures. (K.K. Mathur). 5. The incremental scheduler. (R. Das, J. Saltz, D. Mavriplis and R. Ponnusamy). 6. Automated run-time scheduling of unstructured scientific computation on scalable multiprocessors. (T.-F. Ngai, S. Lundstrom and M. Flynn). 7. Hardware and software architectures for irregular problem architectures. (G.C. Fox). 8. Mapping the adaptive fast multipole algorithm onto MIMD systems. (J.F. Leathrum, Jr. and J.A. Board, Jr.). 9. Massively parallel sparse-matrix computations. (S.G. Kratzer). 10. Semi-structured refinement and parallel domain decomposition methods. (W.D. Gropp and D.E. Keyes). 11. A moving mesh scheme for adaptive domain decomposition. (C.J. Ribbens). 12. A hybrid spectral element-finite difference method for parallel computers. (R. Henderson and G.E. Karniadakis). 13. Some aspects of single-zone structured-grid CFD for a hypercube MIMD computer. (S.R. Chakravarthy and S. Palaniswamy). 14. Mixed finite volume/finite element massively parallel computations: Euler flows, unstructured grids, and upwind approximations. (C. Farhat, L. Fozouli and R. Lanteri). 15. Dynamic mapping of adaptive computations onto linear arrays. (K.R. Pruhs, T.F. Znati and R.G. Melhem). 16. Centralized and distributed dynamic scheduling for adaptive, parallel algorithms. (S.L. Smith and R.B. Schnabel). 17. Random-access bandwidth and grid-based algorithms on massively parallel computers. (W. Celmaster). 18. Performance of PDE sparse solvers on hypercubes. (M. Mu and J.R. Rice). 19. Scan directed load balancing for highly-parallel mesh-connected computers. (E.S. Biagioni and J.F. Prins). 20. Load balancing algorithms on the connection machine and their use in Monte-Carlo methods. (R. Frye and J. Myczkowski).

The Computational Brain. By Patricia S. Churchland and Terrence J. Sejnowski. The MIT Press, Cambridge, Massachusetts and London, England. (1992). 544 pages. \$39.95.

Contents:

1. Introduction. 2. Neuroscience overview. 2.1 Introduction. 2.2 Levels in nervous systems. 2.3 Structure at various levels of organization. 2.4 A short list of brain facts. 3. Computational overview. 3.1 Introduction. 3.2 Looking up the answer. 3.3 Linear associators. 3.4 Constraint satisfaction: Hopfield networks and Boltzmann machines. 3.5 Learning in neural nets. 3.6 Competitive learning. 3.7 Curve fitting. 3.8 Feedforward nets: Two examples. 3.9 Recurrent nets. 3.10 From toy world to real world. 3.11 What good are optimization procedures to neuroscience? 3.12 Models: Realistic and abstract. 3.13 Concluding remarks. 4. Representing the world. 4.1 Introduction. 4.2 Constructing a visual world. 4.3 Thumbnail sketch of the mammalian visual system? 4.4 What is so special about distribution? 4.5 World enough and time. 4.6 Shape from shading: A neurocomputational study. 4.7 Stereo vision. 4.8 Computational models of stereo vision. 4.9 Hyperacuity: From mystery to mechanism. 4.10 Vector averaging. 4.11 Concluding remarks. 5. Plasticity: Cells, circuits, brains, and behavior. 5.1 Introduction. 5.2 Learning and the hippocampus. 5.3 Donald Hebb and synaptic plasticity. 5.4 Memories are made of this: Mechanisms of neuronal plasticity. 5.5 Cells and circuits. 5.6 Decreasing synaptic strength. 5.7 Back to systems and behavior. 5.8 Being and timing. 5.9 Development of nervous systems. 5.10 Modules and networks. 6. Sensorimotor integration. 6.1 Introduction. 6.2 LeechNet. 6.3 Computation and the vestibulo-ocular reflex. 6.4 Time and time again. 6.5 The segmental swimming oscillator. 6.6 Modeling the neuron. 6.7 Concluding remarks. 7. Concluding and beyond. 7.1 Afterword.